Application Level User Traffic Control at the Mobile Network Edge

Evelina Pencheva, Ivaylo Atanasov, Denitsa Velkova, Ventsislav Trifonov
Technical University of Sofia
Sofia, Bulgaria
{enp,iia}@tu-sofia.bg, denitsa.velkova@balkantel.net, vgt@tu-sofia.bg

***Abstract*—Multi-access Edge Computing is a promising technology that may satisfy the diverse quality of service requirements of 5G services. It brings the cloud capabilities for storage and computing in proximity to end users and thus it enables more timely reaction to dynamic network conditions. In this paper, we propose a new mobile edge service which provides functionality for user traffic handling. Using the service interfaces, an intended mobile edge application may shape, redirect or block user traffic based on information about radio conditions, user location, requested data speeds, etc. The new mobile edge service is described by typical use cases, data model, interface definition, and state models which are formally verified.**

I. Introduction

Future fifth generation (5G) networks are expected to improve the performance of mobile networks including quality of service (QoS). The diverse QoS requirements of different services become a challenge for network operators, where the telcos have to take into consideration the limited radio frequency spectrum and the backhaul capacity [1], [2], [3]. The concept of flexible radio access network with cloud intelligence is a viable solution which is a compromise between increasing capacity and total infrastructure costs [4].

Multi-access Edge Computing (MEC) is a technology that provides cloud intelligence at the mobile network edge, distributing computing and storing capabilities. MEC is based on network function virtualization. The virtualization of network functions may cover the control and management of QoS, the service policies and traffic prioritization [5], [6]. It provides flexibility to achieve QoS isolation for customized 5G services through network slicing [7], [8].

According to the MEC technical requirements, the mobile edge platform or intended mobile edge application should provide the possibility for an authorized mobile edge application to statically and/or dynamically register its bandwidth and/or priority requirements, as well as to allocate bandwidth and priority to each session or each application [9].

Bandwidth Management Service (BWMS), defined by ETSI, provides means to efficiently and timely address the requirements of bandwidth and/or priority of various mobile edge applications or sessions of same application [10]. The BWMS Application Programming Interfaces (APIs) provide the possibility for an authorized mobile edge application to allocate bandwidth and/or assign priority to any session or to any application. BWMS can merge all bandwidth management requests and optimize its use.

In this paper, we propose a new mobile edge service that enables dynamic user traffic handling. The research novelty is in providing possibility for mobile edge platform or authorized mobile edge application to inspect, shape, redirect or block user traffic based on policy rules. The traffic handling rules may depend on network congestion, user location, QoS requirements, or may be defined per mobile edge application.

II. Research Motivation

Information on current radio terms is shared through the MEC platform via the Radio Network Information Service (RNIS) service [11]. RNIS is a service that provides radio-related information for mobile edge applications and mobile edge platforms. The granularity of radio network information can be set based on parameters such as cell, user equipment (UE), service quality class, or can be requested over a period of time. The typical information that can be provided is as follows:

- up-to-date information on radio network conditions;
- measurements related to the user plane based on the 3GPP specifications;
- information and its changes related to UEs, served by the radio node(s) connected to the MEC server, including radio resources allocated to UEs.

An authorized mobile edge application may receive a cell level EUTRAN Radio Access Bearer (ERAB) information and may subscribe to receive notifications about RAB establishment, modification and release. Upon receiving a request for RAB management, the mobile edge application is notified.

According to the technical requirements to the MEC among the essential traffic routing properties are the following [9]:

- The MEC platform must provide functionality for routing selected user traffic in the uplink and/or downlink direction to an authorized mobile edge application;
- The MEC platform must provide functionality for inspecting, modifying, shaping user traffic in uplink and/or downlink direction;
- MEC management must allow configuration of traffic rules. Traffic rules can allow the establishment of packet filters based on a network address and / or IP protocol based on Tunnel Endpoint Identifier and/or Subscriber Profile Identifier and/or Quality Class Identifier.

ISSN 2305-7254

The purpose of UE Identity service is to allow UE specific traffic rules in the mobile edge system [12].

UE Identity API provides the functionality for a mobile edge application to register a tag (representing a UE) or a list of tags. The UE Identity tag registration triggers the mobile edge platform to activate the corresponding traffic rule(s) linked to the tag. Later, if the application does not wish to use the traffic rule for that user, it may de-register the UE Identity tag by invoking the de-registration procedure.

The current standard version does not provide a mechanism for realization of tag mapping onto specific UE in the mobile network operator's system. It also does not describe the related application policy information e.g. traffic rule pattern.

In this paper, a new mobile edge service User Traffic Handling Service (UTHS) is considered, based on the MEC routing requirements. The service enables mobile edge applications to manage user traffic. Depending on the policy provided to the mobile edge application's logic, the following actions can be performed on user traffic: continue with existing data, traffic blocking or traffic shaping in uplink and/or downlink directions, as well as user traffic redirection to another destination.

III. Informative Service Description

The proposed mobile edge UTHS allows authorized mobile edge applications to:

- inspect selected uplink and/or downlink user traffic;
- shape selected uplink and/or downlink user plane traffic;
- route selected uplink and/or downlink user traffic from the network to authorized mobile edge application;
- block selected uplink and/or downlink user traffic.

In order to inspect selected user traffic, an authorized mobile edge application needs to subscribe to receive notifications about ERAB management related to specific user with the RNIS. Fig.1 shows the message flow for ERAB setup request approved by an authorized mobile edge application.

A precondition is that the mobile edge application has an active subscription to notifications about ERAB management events. The sequence of steps is as follows:

1) The ERAB setup request is sent by the Mobility Management entity (MME) in the core network.

2) Upon an ERAB setup request, the RNIS service notifies the mobile edge application about the occurrence of the event. The RabEstNotification data structure includes information about the desired ERAB.

3) The mobile edge application confirms the receipt of the notification.

4) Following its logic and information on current radio conditions, the mobile edge application approves the ERAB setup request, sending a POST request to the UTHS service, which includes the RabEstNotification data structure.

5) The mobile edge UTHS sends the original ERAB Setup Request to the eNodeB (eNB) requesting the establishment of the ERABs which will be used for the user traffic.

6) The communication between UE and eNB is based on Radio Resource Control (RRC) protocol. The eNB initiates the RRC Connection Reconfiguration procedure to reconfigure the existing RRC connection.

7) The UE confirms the successful completion of the RRC Connection Reconfiguration.

8) The eNB sends an ERAB Setup Response message confirming the ERAB setup for user traffic.

9) The UTHS sends a response 201 Created to the mobile edge application.

10) An ERAB Setup Response is sent to the MME.

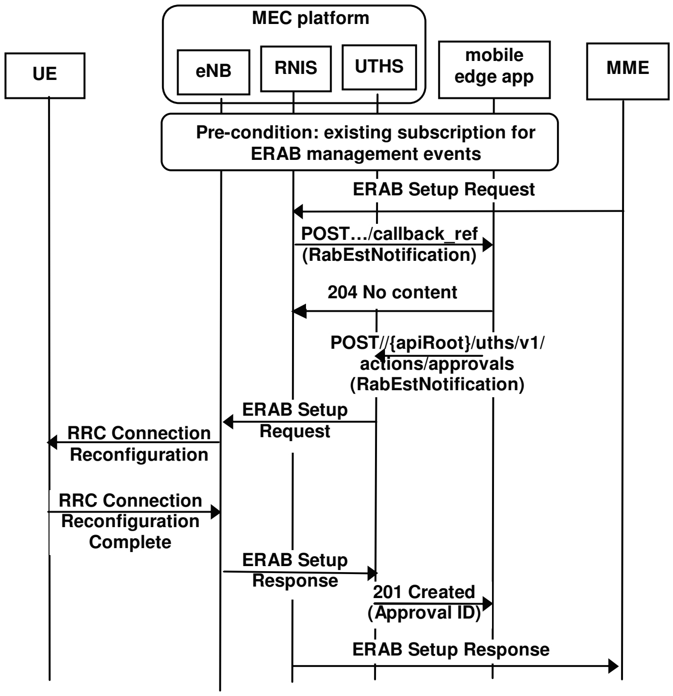

Fig.1. An approval of ERAB setup request by an authorized mobile edge application

Fig.2 shows the message exchange in case of user traffic shaping by an authorized mobile edge application. The sequence of steps is as follows:

1) The mobile edge application sends a POST request for shaping user traffic to UTHS with a body of the message containing the **ShapingInfo** data structure. The **ShapingInfo** data structure defines the user traffic shaping action and filtering criteria.

2) The mobile edge UTHS creates an ERAB Modify Request and sends it to the eNB requesting a change in the radio access bearers dedicated to the user traffic.

3) eNB initiates the RRC Connection Reconfiguration procedure to reconfigure the existing RRC connection.

4) The UE confirms the successful completion of the reconfiguration of the RRC connection.

5) The eNB sends an ERAB Modify Response message, confirming a modification of ERAB dedicated for user traffic.

6) UTHS sends response 201 Created with confirmation of the requested user traffic shaping.

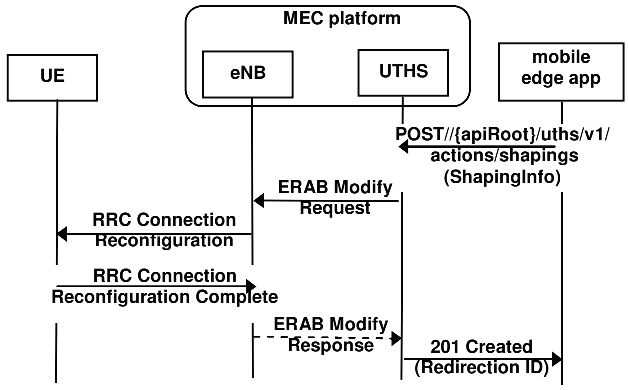

Fig.2. User traffic shaping by an authorized mobile edge application

The message flow when an authorized mobile edge application redirects/blocks the user traffic follows the pattern shown in Fig.2, but the initial request is towards redirections/blockings and the **redirectInfo/gatingInfo** data structure is provided accordingly.

A use case for UTHS service is as follows. An analytic mobile edge application has subscribed to the RNIS service for receiving notifications for ERAB setup, modification, and release dedicated for a specified UE. When an application is notified of the occurrence of an event related to the ERAB setup and/or modification, it may initiate some of the actions related to the user traffic shaping /redirecting /blocking. The reasons for these actions may be information about congestion of the access network, exhaustion of the available user credit, and others.

The proposed UTHS may be used in conjunction with a mobile edge service which provides functionality for usage monitoring control [14]. Usage monitoring control is a part of 3GPP Policy and Charging Control and it enables monitoring and reporting the accumulated usage of network resources on a per session and user basis [15]. This capability is required for enforcing dynamic policy decisions based on the total network usage in real-time.

IV. DATA MODEL

In this section the respective data model is described. The data types related to user traffic handling actions include the following: **ShapingInfo, GatingInfo** and **RedirectionInfo**.

The **ShapingInfo** data type describes bandwidth limitation information. This is a structure of the following items:

- **mBitRateDl** represents the maximum downlink bit rate and it is of Integer type;
- **mBitRateUl** represents the maximum uplink bit rate and it is of Integer type;
- **gBitRateDl** represents the guaranteed downlink bit rate and it is of Integer type;
- **gBitRateUl** represents the guaranteed uplink bit rate and it is of Integer type;
- **TempUeId** is a temporary identifier assigned to a specific UE. This is a structure of **mmec** and **mtmsi**;
- **mmec** is the MME Code, as defined by 3GPP;
- **mtmsi** is the MME Temporary Mobile Subscriber Identity, as defined by 3GPP;
- **shapingDuration** represents the duration of the rate(s) shaping in seconds.

The **BlockingInfo** data type describes information about user traffic blocking. This is a structure of **direction**, **TempUeId** (as described above) and **blockingDuration**, where:

• **direction** indicates the direction of user traffic that should be blocked and it is of Enumeration type: 0 = downlink, 1 = uplink, 2 = downlink and uplink;

• **blockingDuration** represents the duration of the user traffic blocking.

The **RedirectionInfo** data type describes information about user traffic redirecting. This is the structure of **redirectServerAddress**, **TempUeId** (as described above) and **redirectDuration**, where:

• **redirectServerAddress** represents the address of the server to which user traffic must be redirected to;

• **redirectDuration** represents the duration of the application traffic redirection.

The motivation behind the use of the data model is that the different applications store and access same data structures, which enables interoperability.

V. API DEFINITION

The structure of resources, supported by the UTHS, is shown in Fig.3.

Each resource has a unique Uniform Resource Identification (URI). All UTHS resources have the following root:

{apiRoot}/pbwm/{apiVersion}/

Following the RESTful architectural style, all resources are manipulated using four simple operations: CREATE (HTTP POST request), READ (HTTP GET request), UPDATE (HTTP PUT request), and DELETE (HTTP DELETE request. Table I represents the resources and the supported methods.

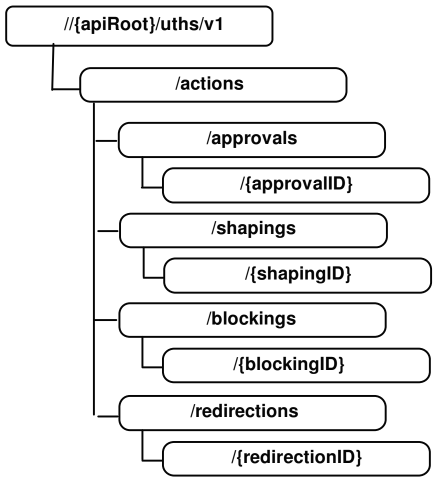

Fig.3 Structure of resources supported by UTHS

TABLE I. RESOURCES AND SUPPORTED HTTP METHODS

Resource name	Resource URI	HTTP method	Description
All actions on user traffic	/actions	GET	Retrieves the list of all enforcement actions.
All approvals of ERAB setup or modification	/actions/ approvals	GET POST	Retrieves a list with all ERAB setup or modification requests approved by the application. Creates a new approval of ERAB setup or modification.
Existing action related to approval of ERAB setup or modification	/actions/ approvals/ approvalID	GET PUT DELETE	Retrieves information about current approval of ERAB setup or modification. Modifies existing approval of ERAB setup or modification. Deletes existing approval of ERAB setup or modification.
All actions related to user traffic shaping	/actions/ shapings	GET POST	Retrieves a list with all actions related to user traffic shaping. Creates a new action related to user traffic shaping.
Existing action related to traffic shaping	/actions/ shapings/ shapingID	GET PUT DELETE	Retrieves information about existing user traffic shaping. Modifies existing user traffic shaping. Deletes existing user traffic shaping.
All actions related to user traffic redirection	/actions/ redirections	GET POST	Retrieves a list with all actions related to user traffic shaping. Creates a new action related to user traffic shaping.
Existing action related to traffic redirection	/actions/ redirections/ redirectionID	GET PUT DELETE	Retrieves information about existing user traffic shaping. Modifies existing user traffic redirection. Deletes existing user traffic redirection.

TABLE I. (CONTINUED)

Resource name	Resource URI	HTTP method	Description
All actions related to user traffic blocking	/actions/ blockings	GET POST	Retrieves a list with all actions related to user traffic blocking. Creates a new action related to user traffic blocking.
Existing action related to traffic blocking	/actions/ blockings/ blockingID	GET PUT DELETE	Retrieves information about existing user traffic blocking. Modifies existing user traffic blocking. Deletes existing user traffic blocking.

VI. STATE MODELS

Deployment of mobile edge UTHS and the respective mobile edge application in the network requires development of models, representing the ERAB states. The models representing the state of ERAB for given UE supported by the MEC platform, by eNB, and by mobile edge application need to be synchronized.

Let us consider an authorized mobile edge application, which subscribes to events related to ERAB setup, modification, and release. When an ERAB management event occurs, the application may approve the ERAB setup/ modification or it may initiate user traffic shaping, redirection, or blocking. The simplified model, representing the mobile edge application view on the ERAB state, is presented in Fig.4.

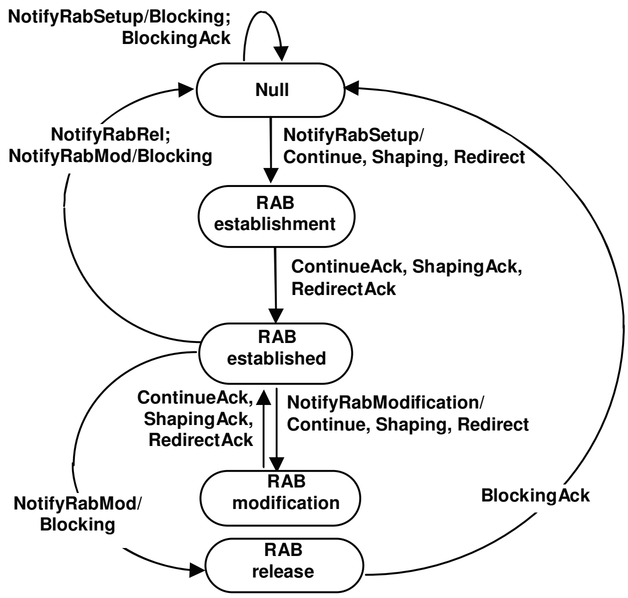

Fig.4. ERAB state model supported by a mobile edge application

The MEC platform supports the API towards the mobile edge applications and the S1AP protocol between MME and eNB [13]. The ERAB setup, modification and release are initiated by the MME in the core network. If the application has an active subscription, in case of ERAB setup /modification request, the MEC platform notifies the application for the event and expects the application instructions how to proceed. In case the mobile edge application approves ERAB setup/modification or initiates user traffic handling actions, the MEC platform composes the corresponding S1AP ERAB management message and sends it to the eNB. The simplified model representing the ERAB state model supported by the MEC platform is presented in Fig.5.

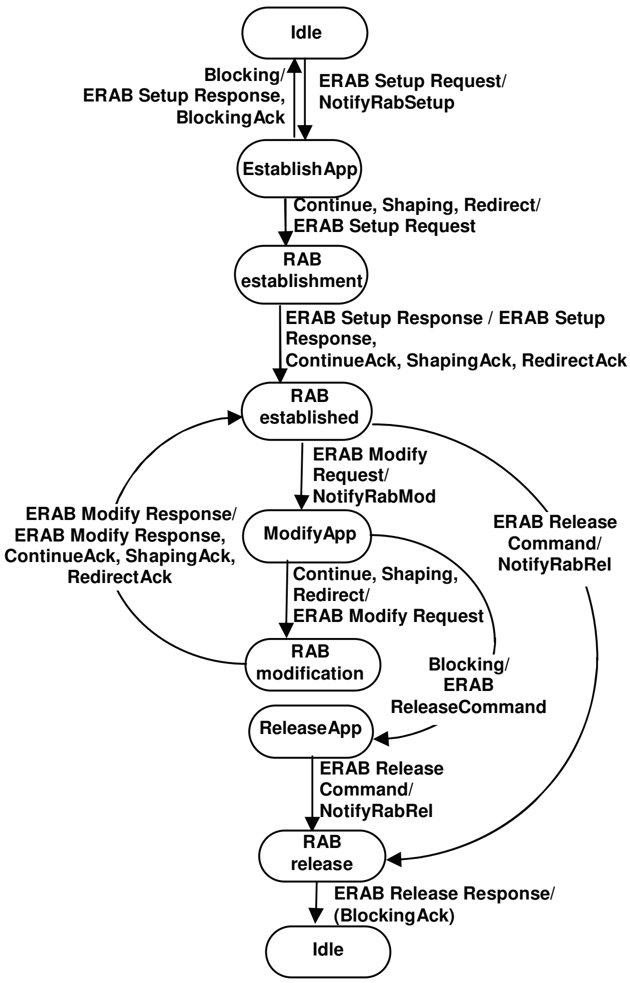

Fig.5. ERAB state model supported by MEC platform

The simplified model representing the view of the eNB of the ERAB state is illustrated in Fig.6. The eNB supports the RRC interface to the UE and the S1AP interface to the MME. The ERAB setup, modification, and release procedures result in reconfiguration or release of the RRC connection.

The three models are simplified as they represent only successful execution of the relevant procedures in the network and do not take into account abnormal conditions associated with unsuccessful completion.

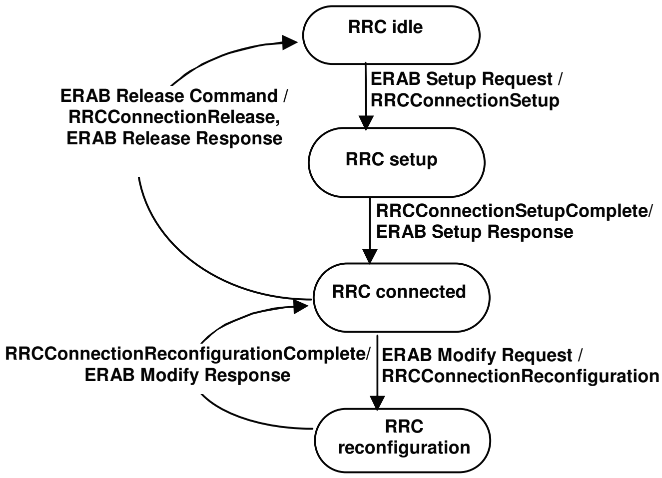

Fig.6. ERAB state model supported by eNB

In order to provide a more rigorous proof that the three models expose equivalent behavior we formalize the modes' descriptions. The description of the ERAB state is formalized using the notion of Labeled Transition System (LTS).

A Labeled Transition System is represented as quadruple of a set of states, a set of actions, a set of transitions, and a set of initial states.

By $T_{App} = (S_{App}, Act_{App}, \rightarrow_{App}, s_0^{App})$ it is denoted an LTS, representing the ERAB state model supported by a mobile edge application, where:

- S_{App} = {Null [s_1^A], RABestablishment [s_2^A], RABestablished [s_3^A], RABmodification [s_4^A], RABrelease [s_5^A]};

- Act_{App} = {NotifyRabSetup[t_1^A], ContinueAck[t_2^A], ShapingAck [t_3^A], RedirectAck[t_4^A], NotifyRabMod [t_5^A], BlockingAck [t_6^A], NotifyRabRel [t_7^A]};

- $\rightarrow_{App} = \{(s_1^A\ t_1^A\ s_1^A), (s_1^A\ t_6^A\ s_1^A), (s_1^A\ t_1^A\ s_2^A), (s_2^A\ t_2^A\ s_3^A), (s_2^A\ t_3^A\ s_3^A), (s_2^A\ t_4^A\ s_3^A), (s_3^A\ t_5^A\ s_4^A), (s_4^A\ t_2^A\ s_3^A), (s_4^A\ t_3^A\ s_3^A), (s_4^A\ t_4^A\ s_3^A), (s_3^A\ t_5^A\ s_5^A), (s_5^A\ t_6^A\ s_1^A), (s_3^A\ t_7^A\ s_1^A), (s_3^A\ t_5^A\ s_1^A)\}$;

$s_0^{App} = \{ s_1^A \}$.

Short notations for states and actions are given in brackets.

By $T_P = (S_P, AcP \rightarrow_{MEC}, s_0^P)$ it is denoted an LTS, representing the ERAB state model supported by MEC platform, where:

- S_P = {Idle [s_1^P], EstablishApp [s_2^P], RABestablishment [s_3^P], RabEstablished[s_4^P], ModifyApp [s_5^P], RABmodification [s_6^P], ReleaseApp [s_7^P], RABrelease [s_8^P]};

- Act_P = { ERABSetupRequest[t_1^P], Blocking [t_2^P], Continue [t_3^P], Shaping [t_4^P], Redirect [t_5^P], ERABSetupResponse [t_6^P], ERABModifyRequest[t_7^P], ERABModifyResponse [t_8^P], ERABReleaseCommand [t_9^P], ERABReleaseResponse [t_{10}^P]};

- $\rightarrow_P$ $= \{(s_1^P\ t_1^P\ s_2^P), (s_2^P\ t_2^P\ s_1^P), (s_2^P\ t_3^P\ s_3^P), (s_2^P\ t_4^P\ s_3^P), (s_2^P\ t_5^P\ s_3^P), (s_3^P\ t_6^P\ s_4^P), (s_4^P\ t_7^P\ s_5^P), (s_5^P\ t_3^P\ s_6^P), (s_5^P\ t_4^P\ s_6^P), (s_5^P\ t_5^P\ s_6^P), (s_6^P\ t_8^P\ s_4^P), (s_4^P\ t_9^P\ s_8^P), (s_5^P\ t_2^P\ s_7^P), (s_7^P\ t_9^P\ s_8^P), (s_8^P\ t_{10}^P\ s_1^P)\}$;

s_0^P $= \{ s_1^P \}$.

By $T_{eNB} = (S_{eNB}, Act_{eNB} \rightarrow_{eNB}, s_0^{eNB})$ it is denoted an LTS, representing the ERAB state model supported by eNB, where:

- S_{eNB} ={ RRCidle[s_1^N], RRCsetup[s_2^N], RRCconnected [s_3^N], RRCreconfiguration [s_4^N]};

- Act_{eNB} = {ERABSetupRequest [t_1^N], RRCConnectionSetupComplete [t_2^N], RABModifyRequest [t_3^N], RRCConnectionReconfigurationComplete [t_4^N], ERABReleaseCommand[t_5^N]};

- $\rightarrow_{eNB} = \{ (s_1^N\ t_1^N\ s_2^N),(s_2^N\ t_2^N\ s_3^N), (s_3^N\ t_3^N\ s_4^N), (s_4^N\ t_4^N\ s_3^N), (s_3^N\ t_5^N\ s_1^N)\}$;

- $s_0^{eNB} = \{ s_1^N \}$.

Having defined formal model descriptions, it may be proved that the three models expose equivalent behavior.

Intuitively, in terms of observed behavior, two LTSs are equivalent if one LTS displays a final result and the other LTS displays the same result. The idea of equivalence is formalized by the concept of bisimilarity [16]. In practice, strong bisimilarity puts strong conditions for equivalence which are not always necessary. The weak bisimilarity admits internal transitions to be ignored.

Proposition: T_{App}, T_P and T_{eNB} are weakly bisimilar.

Proof: As to definition of weak bisimulation, it is necessary to identify a relation between the states of the three LTS, such as for any transition from a state in one LTS there are respective transitions from states in the other LTSs.

By $U_{AppPeNB}$ it is denoted a relation between the states of T_{App}, T_P and T_{eNB}, where $U_{AppPeNB} = \{(s_1^A, s_1^P, s_1^N), (s_3^A, s_4^P, s_3^N)\}$. Then, for each of the following events the following transitions for the states in $U_{AppPeNB}$ are identified:

1. A mobile edge application receives a notification about setup of ERAB dedicated for user traffic and the application approves the ERAB setup: for $(s_1^A\ t_1^A\ s_2^A)$, $(s_2^A\ t_2^A\ s_3^A)$ $\exists \{(s_1^P\ t_1^P\ s_2^P), (s_2^P\ t_3^P\ s_3^P), (s_3^P\ t_6^P\ s_4^P)\} \sqcup \{(s_1^N\ t_1^N\ s_2^N),(s_2^N\ t_2^N\ s_3^N)\}$.
2. A mobile edge application receives a notification about setup of ERAB dedicated for user traffic and the application initiates traffic shaping: for $(s_1^A\ t_1^A\ s_2^A)$, $(s_2^A\ t_3^A\ s_3^A)$ $\exists \{(s_1^P\ t_1^P\ s_2^P), (s_2^P\ t_4^P\ s_3^P), (s_3^P\ t_6^P\ s_4^P)\} \sqcup \{(s_1^N\ t_1^N\ s_2^N),(s_2^N\ t_2^N\ s_3^N)\}$.
3. A mobile edge application receives a notification about setup of ERAB dedicated for user traffic and the application initiates traffic redirection: for $(s_1^A\ t_1^A\ s_2^A)$, $(s_2^A\ t_4^A\ s_3^A)$ $\exists \{(s_1^P\ t_1^P\ s_2^P), (s_2^P\ t_5^P\ s_3^P), (s_3^P\ t_6^P\ s_4^P)\} \sqcup \{(s_1^N\ t_1^N\ s_2^N),(s_2^N\ t_2^N\ s_3^N)\}$.
4. A mobile edge application receives a notification about setup of ERAB dedicated for user traffic and the application initiates traffic blocking: for $(s_1^A\ t_1^A\ s_1^A)$, $(s_1^A\ t_6^A\ s_1^A)$, $\exists (s_1^P\ t_1^P\ s_2^P), (s_2^P\ t_2^P\ s_1^P)$.
5. A mobile edge application receives a notification about modification of ERAB dedicated for user traffic and the application approves the ERAB modification: for $(s_3^A\ t_5^A\ s_4^A)$, $(s_4^A\ t_2^A\ s_3^A)$ $\exists$ { $(s_4^P\ t_7^P\ s_5^P)$, $(s_5^P\ t_3^P\ s_6^P)$, $(s_6^P\ t_8^P\ s_4^P)\} \sqcup \{(s_3^N\ t_3^N\ s_4^N), (s_4^N\ t_4^N\ s_3^N)\}$.
6. A mobile edge application receives a notification about modification of ERAB dedicated for user traffic and the application initiates traffic shaping: for $(s_3^A\ t_5^A\ s_4^A)$, $(s_4^A\ t_3^A\ s_3^A)$ $\exists$ { $(s_4^P\ t_7^P\ s_5^P)$, $(s_5^P\ t_4^P\ s_6^P)$, $(s_6^P\ t_8^P\ s_4^P)\} \sqcup \{(s_3^N\ t_3^N\ s_4^N), (s_4^N\ t_4^N\ s_3^N)\}$.
7. A mobile edge application receives a notification about modification of ERAB dedicated for user traffic and the application initiates traffic redirection: for $(s_3^A\ t_5^A\ s_4^A)$,

$(s_4^A\ t_4^A\ s_3^A) \exists \{(s_4^P\ t_7^P\ s_5^P), (s_5^P\ t_5^P\ s_6^P), (s_6^P\ t_8^P\ s_4^P)\} \sqcup \{(s_3^N\ t_3^N\ s_4^N), (s_4^N\ t_4^N\ s_3^N)\}$.

8. A mobile edge application receives a notification about modification of ERAB dedicated for user traffic and the application initiates traffic blocking: for $(s_3^A\ t_5^A\ s_1^A)$, $(s_1^A\ t_6^A\ s_1^A) \exists \{(s_4^P\ t_7^P\ s_5^P), (s_5^P\ t_2^P\ s_7^P), (s_7^P\ t_9^P\ s_8^P), (s_8^P\ t_{10}^P\ s_1^P)\} \sqcup (s_3^N\ t_5^N\ s_1^N)\}$.

9. A mobile edge application receives a notification about release of ERAB dedicated for user traffic: for $(s_3^A\ t_7^A\ s_1^A)$ $\exists \{(s_4^P\ t_9^P\ s_8^P), (s_8^P\ t_{10}^P\ s_1^P)\} \sqcup (s_3^N\ t_5^N\ s_1^N)$.

Therefore T_{App}, T_{P} and T_{eNB} are weakly bisimilar, i.e. they expose equivalent behavior. ■

VII. CONCLUSION

5G network infrastructure will exploit network function virtualization to provide adequate quality of service and traffic prioritization based on service policies. Multi-access Edge Computing possesses the potential to fulfill the diverse QoS requirement of the emerging 5G applications and to improve the Quality of Experience for the end users.

Proactive dynamic user traffic handling, including QoS control and management, can be realized with a new mobile edge service that provides functionality for user traffic shaping, redirecting and blocking. In this paper, we propose a new mobile edge service as a response to the MEC traffic routing requirements. The service is described by means of typical use cases, data model, API definition, and state models.

Centralized control of user traffic, applied in the vicinity of end-users, is expected to have a positive impact on network performance due to the effectiveness of coordination and exposure of bandwidth information in the radio access network. The uplink user traffic does not need to pass through the network core as policy-based decision can be made in the radio access network saving backhaul and transmission resources.

ACKNOWLEDGMENT

The research is conducted under the grant of project DH07/10-2016, funded by Bulgarian National Science Fund, Ministry of Education and Science.

REFERENCES

[1] T. Shuminoski, L. Gavrilovska and T. Janevski, "QoS performances of heterogeneous networks with multiple radio access technologies," *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, Ohrid, 2017, pp. 132-137.

[2] R. A.Aljiznawi, N. H. Alkhazaali, S. Q. Jabbar, .D. J. Kadhim, "Quality of Service (QoS) for 5G, " *International Journal of Future Computer and Communication*, vol. 6, no. 1, March 2017, pp.27-30.

[3] S. Guergov. Acupressure in magneto therapy environment, *Series on Biomechanics*, Bulgarian Academy of Science, vol.32, No.1, 2018, p. 16-19.

[4] J. Pedreno-Manresa, P. S. Khodashenas, M. S. Siddiqui and P. Pavon-Marino, "Dynamic QoS/QoE assurance in realistic NFV-enabled 5G Access Networks," *2017 19th International Conference on Transparent Optical Networks (ICTON)*, Girona, 2017, pp. 1-4.

[5] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, T. Taleb, "Survey on Multi-Access Edge Computing for Internet of Things Realization", *IEEE Communications Surveys & Tutorials*, 2018, doi: 10.1109/COMST.2018.2849509

[6] R.Bolla, A. Carrega, M. Repetto, G. Robino, "Improving Efficiency of Edge Computing Infrastructures through Orchestration Models," *Computers*, 2018, vol.7 issue 36, pp.1-17.

[7] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen and X. Li, "End-to-End Quality of Service in 5G Networks: Examining the Effectiveness of a Network Slicing Framework," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 65-74, June 2018.

[8] Y. Liu, G. Shou, Y. Liu, Y. Hu, Z. Guo, "Performance Evaluation of Integrated Multi-Access Edge Computing and Fiber-Wireless Access Networks, " *IEEE Access*, 2018, vol.6, pp. 30269-30-279.

[9] ETSI GS MEC 002. *Mobile Edge Computing (MEC); Technical Requirements*, v1.1.1, 2016

[10] ETSI GS MEC 015. *Mobile Edge Computing (MEC); Bandwidth Management API*, v1.1.1, 2017.

[11] ETSI GS MEC 012, *Mobile Edge Computing (MEC); Radio Network Information API*, v1.1.1, 2017.

[12] ETSI GS MEC 014. *Mobile Edge Computing (MEC); UE Identity API*, v1.1.1, 2017

[13] 3GPP TS 36.331 (2018). Evolved Universal Terrestrial Radio Access (EUTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP); Release 14, v14.1.0.

[14] E Pencheva, I. Atanasov, Usage Monitoring Control in Radio Access Network, *23th Conference of Open Innovations Association FRUCT*, Bologna, Italy, 2018, pp.1-9.

[15] 3GPP TS 23.203 Technical Specification Group Services and System Aspects; Policy and Charging Control architecture, 2018, Release 15, v15.3.0.

[16] X. J. Chen, R. De Nicola, "Algebraic characteristics of trace and decorated trace equivalences over tree-like structures," *Theoretical Computer Science*, 254, Elsevier, 2001, pp.337-361.